WBS 4.2-2

KNOWLEDGE BASE MANAGEMENT SYSTEMS STUDY

REPORT #1

made subject to [illegible]. The limitations contained in this legend will be considered void after September 28, 1988 [illegible] shall be marked on any reproduction of this data in whole or in part.

Prepared under Contract No. NAS1-17555 by
BOEING COMMERCIAL AIRPLANE COMPANY
P.O. Box 3707
Seattle, Washington 98124

for

Langley Research Center
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Prepared by:

L. S. Baum — Date: 9/28/84

W. J. McClay — Date: 9/28/84

S. J. Lee — Date:

Responsible Manager:

H. R. Johnson — Date: 9/28/84

Approved by:

W. A. Bryant — Date:

KNOWLEDGE BASE MANAGEMENT SYSTEMS STUDY
(REPORT #1)
TABLE OF CONTENTS

KNOWLEDGE BASE MANAGEMENT SYSTEMS STUDY
(REPORT #1)
TABLE OF CONTENTS

1.0 INTRODUCTION

IPAD has begun research into the development of Knowledge Base Management (KBM) technology, with emphasis on the feasibility and desirability of building an integrated Knowledge/Data Base Management System and on the application of such technology to future computer-aided engineering systems.

1.1 PURPOSE OF THIS DOCUMENT

The purpose of this document is to survey current research and industrial development of Artificial Intelligence applications and to formulate directions for knowledge base technology research.

1.2 ARTIFICIAL INTELLIGENCE OVERVIEW

Artificial Intelligence (AI) is the discipline of computer science that strives to make computers more useful by imbuing systems with characteristics considered intelligent. The need for more intelligent processing and advancements in hardware and software have led to renewed interest in AI. This, together with some 25 years of AI foundation, leads to visions of real payoffs in AI applications. This overview briefly defines AI, describes its scope, identifies key technical issues, and describes some applications.

Defining AI is not an easy task. Elaine Rich defines AI by putting a boundary around the concept. She states, "I propose the following is by no means a universally- accepted definition." AI "is the study of how to make computers do things at which, at the moment, people are better." [Rich 83] This definition shows us that part of the problem with pinning down the meaning of AI is that the field is still emerging. The boundaries of AI also tend to be indefinite because concensus among principals in the field has not been reached. Another way to define AI is to describe a test for AI. Alan M. Turing's idea was to put a human in one room and a machine in another and allow a tester to communicate with the rooms via a teletype. If the tester could not tell which room contained the human and which contained the machine, the machine would be deemed intelligent [Hofstad 80].

The complex nature of AI requires a large amount of formally-encoded knowledge and a means for interpreting it. The areas where AI are useful are broad and varied. The following list [Rich 83] provides a summary of the research areas that fall within the scope of artificial intelligence:

- game playing
- theorem proving
- general problem solving
- perception
  - vision
  - speech
- natural language understanding

- expert problem solving
  - symbolic mathematics
  - medical diagnosis
  - chemical analysis
  - engineering design

Applications of AI technology will be used in farms, mining operations, factories, offices, schools, hospitals, and households.

Technical AI issues include efficient search algorithms, appropriate applications of AI techniques, methods of representing knowledge, controlling machine attention and appropriate programming languages. Some problems require such large knowledge bases that solutions could take hundreds of years on our fastest computer using traditional search techniques. Management of these large knowledge bases is a prime issue of AI.

Expert system technology is receiving the greatest amount of attention within the AI community. There have been many successful applications. Because much of engineering is based on principles and rules, expert systems are suitable for engineering applications [Gevater 83]. AI applications currently coming into use will lead the way toward truly intelligent systems.

1.3 KNOWLEDGE BASE OVERVIEW

During the past ten years, data base technology has reached the marketplace in the form of many successful DBMS systems. The general purpose DBMS has provided an important tool which has allowed data to be defined outside of application programs. This allows data to be defined uniformly among many applications within a large system and provides a method of applying uniform security restrictions preventing unauthorized users from accessing sensitive data. Other benefits of DBMS technology include transaction management, less redundancy, greater consistency, views, concurrent access, automatic backup and recovery and very sophisticated query facilities which allow powerful ad hoc report and graphics capabilities.

Meanwhile, Artificial Intelligence research has proceeded without concern for the commercialization of their results. This research has shown that various types of knowledge within programs can be captured essentially as data. State-of-the-art AI applications make use of advanced techniques for organizing this knowledge/data, but for efficiency reasons, implementors have not attempted to separate knowledge from these applications. Such separation would allow other applications to share the knowledge, thus eliminating possible duplication. Indeed security, automatic recovery, concurrency, views, ad hoc query, and other data management functions, which are routinely provided in present day DBMS systems, should be provided in future KBMS systems.

It seems highly probable that in the future, users will have the need for security, data/knowledge sharing and other features as knowledge base management technology finds its way to the

marketplace. Just as data base management matured in the process of developing a general purpose tool which ushered in the era of data independent applications, knowledge base management will mature in order to meet the ever growing list of demands for "industrial strength" software systems which emphasize maintainability, reliability, and consistency.

At this juncture, it appears that we have an opportunity to advance knowledge base management technology from the ivory tower environment to the real world by exploiting data base management technology. This could be accomplished either by commercial software writers or by AI research groups, but more likely will be a joint effort utilizing the best elements of both worlds.

2.0 ARTIFICIAL INTELLIGENCE TECHNOLOGY AND RESEARCH

2.1 EXPERT SYSTEMS

Feigenbaum [Feig 82] defines an expert system as:

> An intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human experience for their solution.

An expert system is composed of the following:

1) A KNOWLEDGE BASE of facts, rules, and heuristics that represent an expert's knowledge of the problem. Most current systems are "rule based"; i.e., their knowledge base consists of 'production rules' which are statements of the form: IF condition THEN action.

2) A GLOBAL DATA BASE that acts as the working memory for the particular problem currently being analyzed.

3) An INFERENCE MECHANISM. Applicable items in the knowledge base are applied to the facts in the global data base to draw inferences. These inferences result in new items to be stored in the knowledge base or new facts for the data base.

In addition, it is desirable that expert systems include:

4) A user-friendly NATURAL LANGUAGE and/or SYMBOLIC INTERFACE to facilitate the stating of a particular problem to be solved and the transfer of knowledge from the human expert.

5) An EXPLANATION MODULE which explains the reasoning behind the chosen solution. This serves both as a debugging aid and training tool.

2.1.1 Current Applications

The development of expert systems has increased dramatically recently, with applications in an extremely large variety of areas. Figure 2.1-1 [Gevater 83] demonstrates the degree to which expert system technology is being exploited in this country. As can be seen from the list, a large percentage of this work is concentrated in the fields of medicine, chemistry, and geology, however, the list also indicates the extent to which expert systems development is occurring in other fields as well.

As impressive as the list in Figure 2.1-1 is, it should be noted, that there are several limitations on what can be achieved with expert systems at the current time. No existing expert system has expertise beyond a narrow domain of knowledge; i.e., an expert

system might have expertise in medicine or organic chemistry, but the creation of a system with expertise in the natural sciences in general is beyond the scope of expert system technology today. Expert systems have extremely fragile behavior at the 'boundaries' of their expertise; whereas, a human expert's performance will normally degrade gracefully when the limits of his or her expertise is reached. Furthermore, until much more progress is made in the area of natural language understanding, expert systems will remain very limited in their ability to learn from mistakes and to add, modify, and delete data and rules automatically.

Function	Domain	System*	Institution
Diagnosis	Medicine	PIP	M.I.T.
	Medicine	CASNET	Rutgers U.
	Medicine	INTERNIST/CADUCEUS	U. of Pittsburgh
	Medicine	MYCIN	Stanford U.
	Medicine	PUFF	Stanford U.
	Medicine	MDX	Ohio State U.
	Computer Faults	DART	Stanford U./IBM
	Computer Faults	IDT	DEC
	Nuclear Reactor Accidents	REACTOR	E G & G Idaho Inc.
Data Analysis and Interpretation	Geology	DIPMETER ADVISOR	M.I.T./Schlumberger
	Chemistry	DENDRAL	Stanford U.
	Chemistry	GA1	Stanford U.
	Geology	PROSPECTOR	SRI
	Protein Crystallography	CRYSALIS	Stanford U.
	Determination of Causal Relationships in Medicine	RX	Stanford U.
	Determination of Causal Relationships in Medicine	ABEL	M.I.T.
	Oil Well Logs	ELAS	AMOCO
Analysis	Electrical Circuits	EL	M.I.T.
	Symbolic Mathematics	MACSYMA	M.I.T.
	Mechanics Problems	MECHO	Edinburgh
	Naval Task Force Threat Analysis	TECH	Rand/NOSC
	Earthquake Damage Assessment for Structures	SPERIL	Purdue U.
	Digital Circuits	CRITTER	Rutgers U.
Design	Computer System Configurations	R1/XCON	C.M.U./DEC
	Circuit Synthesis	SYN	M.I.T.
	Chemical Synthesis	SYNCHEM	SUNY Stonybrook

FIGURE 2.1-1--EXISTING EXPERT SYSTEMS BY FUNCTION

Function	Domain	System*	Institution
Planning	Chemical Synthesis	SECHS	U. of Cal. Santa Cruz
	Robotics	NOAH	SRI
	Robotics	ABSTRIPS	SRI
	Planetary Flybys	DEVISER	JPL
	Errand Planning	OP-PLANNER	Rand
	Molecular Genetics	MOLGEN	Stanford U.
	Mission Planning	KNOBS	MITRE
	Job Shop Scheduling	ISIS-II	CMU
	Design of Molecular Genetics Experiments	SPEX	Stanford U.
	Medical Diagnosis	HODGKINS	M.I.T.
	Naval Aircraft Ops	AIRPLAN	CMU
	Tactical Targeting	TATR	RAND
Learning from Experience	Chemistry	METADENDRAL	Stanford U.
	Heuristics	EURISKO	Stanford U.
Concept Formation	Mathematics	AM	CMU
Signal Interpretation	Speech Understanding	HEARSAY II	CMU
	Speech Understanding	HARPY	CMU
	Machine Acoustics	SU/X	Stanford U.
	Ocean Surveillance	HASP	System Controls Inc.
	Sensors On Board Naval Vessels	STAMMER-2	NOSC, San Diego/SDC
	Medicine—Left Ventrical Performance	ALVEN	U. of Toronto
	Military Situation Determination	ANALYST	MITRE
Monitoring	Patient Respiration	VM	Stanford U.
Use Advisor	Structural Analysis Computer Program	SACON	Stanford U.
Computer Aided Instruction	Electronic Troubleshooting	SOPHIE	B.B.N.
	Medical Diagnosis	GUIDON	Stanford U.
	Mathematics	EXCHECK	Stanford U.
	Steam Propulsion Plant Operation	STEAMER	BBN
	Diagnostic Skills	BUGGY	BBN
	Causes of Rainfall	WHY	BBN
	Coaching of a Game	WEST	BBN
	Coaching of a Game	WUMPUS	M.I.T.
		SCHOLAR	BBN

FIGURE 2.1-1--EXISTING EXPERT SYSTEMS BY FUNCTIONS (CONT.)

Function	Domain	System*	Institution
Knowledge Acquisition	Medical Diagnosis	TEIRESIAS	Stanford U.
	Medical Consultation	EXPERT	Rutgers
	Geology	KAS	SRI
Expert System Construction		ROSIE	Rand
		AGE	Stanford U.
		HEARSAY III	USC/ISI
		EMYCIN	Stanford U.
		OPS 5	CMU
		RAINBOW	IBM
	Medical Diagnosis	KMS	U. of MD
	Medical Consultation	EXPERT	Rutgers
	Electronic Systems Diagnosis	ARBY	Smart Sys. Tech.
	Medical Consultation Using Time-Oriented Data	MECS-AI	Tokyo U.
Consultation/Intelligent Assistant	Battlefield Weapons Assignments	BATTLE	NRL AI Lab
	Medicine	Digitalis Therapy Advisor	M.I.T.
	Radiology	RAYDEX	Rutgers U.
	Computer Sales	XCEL	CMU/DEC
	Medical Treatment	ONCOCIN	Stanford U.
	Nuclear Power Plants	CSA Model-Based Nuclear Power Plant Consultant	GA Tech
	Diagnostic Prompting in Medicine	RECONSIDER	U. of CA, S.F.
Management	Automated Factory	IMS	CMU
	Project Management	CALLISTO	DEC
Automatic Programming	Modelling of Oil Well Logs	ΦNIX	Schlumberger-Doll Res.
		CHI	Kestrel Inst.
		PECOS	Stanford U.
		LIBRA	Stanford U.
		SAFE	USC/ISI
		DEDALUS	SRI
		Programmer's Apprentice	M.I.T.
Image Understanding		VISIONS	U. of Mass.
		ACRONYM	Stanford U.

FIGURE 2.1-1 EXISTING EXPERT SYSTEMS BY FUNCTION (CONT.)

2.1.2 Engineering Applications

Engineers are becoming more and more aware of the potential benefits of expert systems technology in their work. A large number of expert systems have been or are being developed to solve engineering problems.

There is considerable work being done at The Boeing Company to develop expert systems for various engineering applications; some of these projects are described below. These projects are being developed using expert systems building tools such as KS300 and OPS5.

2.1.2.1 Optimization

In general, to search for an "optimal solution" is too impractical and time-consuming, even for a computer. Instead, expert systems use heuristics to determine one or more near-optimal solutions. This technique is important for systems such as SYN (circuit sythesis) and the CSA Nuclear Power Plant Consultant.

2.1.2.2 Design Tools

SYN aids in the design of electronic circuits. At The Boeing Company, a Composites Advisor is being developed that will assist in the selection of resins and fibers for composite parts. Boeing is also developing a Computer Service Selector to assist in choosing appropriate computer services for an engineering task.

2.1.2.3 Real Time Management

The nuclear power plant accident diagnostician, REACTOR, must perform in real time. This type of application is important, not because a human expert is unavailable or too expensive, but because the short reaction time may cause a human to make an error in judgement which could cause extensive damage or even loss of life.

2.1.2.4 Planning

At Boeing, several expert systems are being developed to assist in various planning activities. The Machinability Expert System is designed to increase the effectiveness of the BAC Machinability Model in the manufacturing planning process. BMAC is developing the Multiple Attack Planning Assistant, as well as the Crew Station Information Manager. BCAC is working on an expert system called TOOLBOX to assist in the selection of expert system building tools for CAD/CAM applications.

2.2 OTHER ARTIFICIAL INTELLIGENCE APPLICATIONS

2.2.1 Computer Vision

Computer vision is perception of a scene based upon visual sensory input. The input consists of two-dimensional arrays of brightness values, color vectors, or similar digitized images. Artificial intelligence techniques are used to accomplish perception, that is, the assignment of meaning to the images described by these arrays.

In general, meaning is achieved by matching images within a scene with stored models. There are several approaches toward that end.

In the Hierarchical Bottom-Up Approach, features are extracted from the image via line and area determination techniques, the features are then represented by predefined symbols, and the scene is interpreted using models of how these symbols can be related. This approach works well where there are simple scenes with relatively few previously defined symbols but is inappropriate for a general-purpose computer vision system.

In the Hierarchical Top-Down Approach, heuristics are used to generate test models, which are compared with the features found in the scene. While this approach is better suited than the Bottom-Up approach for interpreting complicated scenes, such systems are not easily modified to deal with new applications.

The Heterarchical Approach uses a central monitor to oversee the performance of the system. If the test models being generated are not being successfully matched, the monitor may modify the generation algorithms. Similarly, the monitor can alter the feature generation component to maximize the succesful discovery of relevant lines and areas in the image.

In the Blackboard Approach, the model generation, feature extraction, and symbolic representation modules each run independently, sharing data via an area called the blackboard. This approach is particularly attractive in cases where several hypotheses must be considered simultaneously and has a natural fit with parallel processing capabilities.

2.2.2 Natural Language Processing

This area of research is concerned with computer comprehension of natural language including printed matter, as opposed to programming languages in various forms. A general-purpose natural language processing capability requires a vast amount of knowledge including the meaning of words, sentence structure and grammar, contextual knowledge and general knowledge of the world. Figure 2.2-1 [Gevarter 83] presents a list of currently available commercial natural language systems.

2.2.3 Speech Recognition

This area differs from natural language processing in that its goal is to determine what words were spoken--where the input is a continual stream of phonemes, rather than to determine the meaning of what was said. This involves pattern matching against a set of patterns representing the patterns of the words in the reference dictionary.

Speech recognition poses many problems. Even if the words to be recognized come from the same individual, there will be sharp variations in the acoustic signal from one occurrence of the word to the next. Thus, the signal must be modified to find a match in the reference dictionary. This problem is greatly exacerbated when additional speakers must be understood. Other extremely difficult problems include detecting the divisions between words and recovering a phoneme that is lost when it is both the terminal phoneme of a word and the initial phoneme of the next word. All of these problems become much more difficult when real time response is required.

System	Organization	Purpose	Comments
INTELLECT (Derivative of ROBOT) $50K/System (also distributed as ON-LINE ENGLISH and GRS Executive)	Artificial Intelligence Corp. Waltham, Mass (Culliane) (Information Sciences)	NLI for Data Base Retrieval (Other extensions underway)	• Several hundred systems sold • Takes about 2 weeks to implement for a new data base. • Written in PL-1 • Available for mainframes
PEARL (Based on SAM and PAM) $250K/system	Cognitive Systems New Haven, Conn	Custom NLI's The first system—Explorer—is an interface to an existing map generating system. Others are interfaces to data bases.	• Large start-up cost in building the knowledge base. • Several systems have been, and are being, built. • Written in LISP
Straight Talk (Derivative of LIFER) $660	Dictaphone, Written by Symantec Sunnyvale, CA	Highly portable NLI for DBMS for micro-computers.	• Written in PASCAL. Designed to be very compact and efficient. Available about Nov. 1983. • User customized.
SAVVY $950	SAVVY Marketing International Sunnyvale, CA	System Interface for micro-computers	• Not linguistic. Uses adaptive (best fit) pattern matching to strings of characters. • Released 3/82 • User customized
Weidner System $16K/language direction	Weidner Communications Corp. Provo, UT	Semi-Automatic Natural Language Translation.	• Linguistic approach. Written in FORTRAN IV. • Translation with human editing is approximately 100 words/hr (up to eight times as fast as human alone). • Approx. 20 sold by end of 1982, mainly to large multi-national corporations.
ALPS	ALPS Provo, UT	Interactive Natural Language Translation	• Linguistic Approach • Uses a dictionary that provides the various translations for technical words as a display to human translator, who then selects among the displayed words.

FIGURE 2.2-1--SOME COMMERCIAL NATURAL LANGUAGE SYSTEMS

System	Organization	Purpose	Comments
NLMENU	Texas Instruments, Inc. Dallas, TX	NLI to Relational Data Bases	• Menu Driven NL Query System • All queries constructed from menu fall within linguistic and conceptual coverage of the system. Therefore, all queries entered are successful. • Grammars used are semantic grammars written in a context-free grammar formalism. • Producing an interface to any arbitrary set of relations is automated and only requires a 15-30 minute interaction with someone knowledgeable about the relations in question. • System will be available late in 1983 as a software package for a

FIGURE 2.2-1--SOME COMMERCIAL NATURAL LANGUAGE SYSTEMS (CONT.)

2.3 KNOWLEDGE REPRESENTAION

A knowledge representation is a data structure coupled with an interpretive program (inference engine) which can intelligently manipulate the data structure and draw inferences from its content. Procedures may be attached to the data structure in which case, the inference engine may initiate execution of these procedures.

2.3.1 Production Rules

Production rule-based systems organize knowledge into production rules which are simply condition-action pairs. They take the form of "IF condition A is true, THEN take action B." For example, an expert football coach's rule might be "IF it is fourth down and long yardage is required, THEN punt." Such systems are used to represent knowledge about how people do a specific job such as medical diagnosis or mineral exploration. The "IF ... THEN ..." nature of these rules is a fairly natural form for humans to express their knowledge. However, reformulating an expert's set of rules and using them to build an expert system requires much time and skill to insure that the rules adequately describe the decision-making process within the problem domain.

The inference mechanism scans the production rules looking for those which are applicable in the current context. It is possible that more than one rule may apply. Consider the football coach example above. IF it is fourth down and twenty yards are required for a first down, THEN the system may readily recommend that the team should punt. But what if we add a rule which says "IF it is fourth down and long yardage is required and there is less than one minute to play, THEN pass." The inference mechanism would find that during the last minute of play there are two rules which apply and would have made the smarter choice: recommend passing. The system could determine which rule to apply by considering the order in which the rules appear in the knowledge base. Using the first rule in the list which applies to the current situation would force the knowledge engineer to arrange the rules in the order of importance causing an unnatural burden. The human thought mechanism does not require such orderings; it simply applies the "most reasonable" rule. More advanced inference mechanisms allow any ordering and resolve such conflicts in imaginative ways but generally incur increased complexity of rule expression or restrictions on the set of allowable semantic primitives.

Some production systems can handle complex problems which involve generation and evaluation of a number of intermediate steps before reaching an objective. An example of such a problem is finding the cheapest air fare between two cities, where stopovers are allowed. Every combination of flights which lead to the destination must be considered. The addition of a rule stipulating the maximum layover time for connecting flights can make the problem even more interesting. Problems such as these may utilize a backward search, first considering all the flights coming into the destination city then working backwards, although greater efficiently is often possible when a combination of forward and backward searching is applied.

Some problems are not tractable and result in combinatorially explosive searches. The human approach to such problems is to find a solution which may not be optimum but does represent careful consideration of the factors and is probably close to optimum. A representative problem is that of determining the most efficient route for the salesman to make his calls which are scattered over some geographic area. Rules could be constructed regarding the time required to travel between points, when he can see certain people, and his desire to end up at a point close to home. The number of possible itineraries could be very large and determing that an "adequate" solution has been reached is not easy. Solving this and other problems which are considerably more complex challenges the knowledge engineer to write rules which concisely express the problem domain. This job can be made considerably easier, however, by a more powerful rule taxonomy and/or a more intelligent inference mechanism.

2.3.2 Logic

Logic systems organize knowledge within the precise framework of mathematical logic where we need only be concerned with the truth or falsity of expressions. This may not provide the richest possible structure for expressing a given problem domain, but the rules of inference which are used to answer questions about the domain are mathematically sound and complete.

The logic formalism used most often in AI is first-order logic. This is based on predicate calculus which is in turn based on propositional calculus. First-order logic embodies most of the notions of propositional calculus (items 1-5 below) and predicate calculus (items 6-8 below) but is made less general and more succint by the addition of a few rules (items 9-11 below). The elements of first order logic include the following:

1) Constants.
2) Variables which are simply placeholders for some constant.
3) Sentential connectives such as "and", "or", "not", "implies" and "equivalent".
4) Propositions such as "George is the father of Mary".
5) The "modus ponens" rule which says that if X is true and X implies Y, then Y is true.
6) Predicates, which are statements about objects which when applied to specific objects have a value of either true or false.
7) The existential quantifier and the universal quantifier.
8) Inference rules for quantifiers (e.g., If you have the proposition "For all X, IF X is a man, THEN X is a mortal" then you have "John is a man" provided you can show that John is a mortal.
9) Functions, which are similar to predicates, but can return objects instead of just true or false.
10) The predicate "equals".
11) Quantification over individuals but not over predicates and functions.

First-order logic is a monotonic knowledge representation. A fact can either be true or false and it cannot be true one minute and then false the next. Furthermore, for any given set of facts, conclusions implied by those facts cannot be falsified by the addition of any new fact.

Languages such as PROLOG are based on first-order logic but are more or less efficient depending upon the resolution procedure used for inferencing. These systems have many of the same problems as some production rule systems. The ordering of statements in PROLOG, for instance, is important to the inferencing process. This causes a PROLOG program to be less modular than a pure first-order logic statement of the problem.

Other factors include whether the system must find all possible solutions or can be custom tailored to solve some particular class of problem. Many AI systems, even some production rule expert systems, used first-order logic as the basis of their inference mechanism.

In addition to the implementation problems of efficiency and representation, there is also the fact that first-order logic is not always the most natural way of expressing knowledge. Furthermore, it is not even claimed by its most staunch supporters that everything is representable in this form.

Another form of logic is called nonmonotonic logic. Our own common sense is nonmonotonic because it allows us to consider propositions to be true until such time that they lead to logical conflicts. Then we either cast them out or make changes which allow them to remain compatible with our beliefs. First-order logic is strictly monotonic, but future systems will probably augment first-order logic with some yet to be determined rules of reason to guide the inferencing process which may perform in some ways like common sense. In spite of all the controvery, it is widely accepted that logic will undoubtedly play an important role in future AI systems.

2.3.3 Semantic Nets

A semantic network consists of nodes and arcs. The nodes usually represent objects and the arcs represent connecting relationships between the objects.

Example:

Generic:	Node 1	Arc 1,2	Node 2
Specific:	Pat	Likes	Mike
	Pat	Runs	Races
	Likes	Is-a	Feeling
	Is-a	Occurs	Often

Searching through knowledge encoded in a semantic network is usually done using pattern matching. By referencing the above example, one can see that a search of the knowledge network could supply information of the form: Pat likes Mike, Pat and Mike's relationship is based on feeling. The IS-A relationship is a common relationship, and Pat does other things beside liking Mike.

One advantage of semantic nets is that they have a minimal degree of complexity. That makes the knowledge easy to maintain and comps. First-order logic is strictllicated concepts can be formed by adding as many related node pairs as required. This simplicity of knowledge representation leads directly to the main disadvantages of a semantic net, which are the lack of a standard way of representing a specific fact and the high overhead for searching through large networks.

Partitioned networks and conceptual dependencies are network knowledge models that are more powerful than simple semantic networks. A partitioned network is a grouping of semantic primitives or a structuring of node/arc relationships. By grouping, more knowledge can be retrieved with a single access. More semantics are now contained in the representation. The knowledge added via the structure is not just primitive; for now, the system can use the meaning carried by the structure. This added power is not free as some flexibility is lost. Conceptual dependency is a structure that is specialized. It is used to represent the components of an active environment. Networks can become very specialized and structured to the extent that their identity as a network can become almost lost. This is shown in the next section.

2.3.4 Frames and Scripts

Frames and scripts are very powerful examples of structured semantic nets. They have become very popular because they are felt to closely parallel the way humans conceptualize complex objects or common sequences of events. A frame is a semantic structure composed of slots. In its simplest form, it could be thought of as a record composed of fields. Slots may also be frames or pointers to other frames. Slots are ultimately assigned symbolic primitives, which are sometimes called facets. A frame for a rectangular parallelipiped might have twelve edge slots, eight corner slots, six side slots, and a volume slot. A corner slot would point to a corner frame, that contains a vertex and three angles. An application using this frame may have all the information for the object except the volume. If the volume were needed, a procedure would be triggered by the frame logic and the volume information computed for the application. This is an example of procedural attachment, which is another useful feature that is commonly applied when using frames.

Scripts are used to represent sterotypical sequences of events, for example, a trip to a restaurant. Scripts superficially look very much like frames in that they are composed of slots. However, a typical script is composed of the following components as typified by the restaurant script [Rich 83]:

COMPONENT	DESCRIPTION
Entry condition-	be hungry and have money
Result-	has less money and is not hungry
Props-	menu, table, and chairs
Roles-	customer, waiter, and cook
Track-	menu on table or waiter brings menu or customer asks for menu
Scenes-	entering, ordering, eating and exiting

With a well-defined script for a restaurant and the knowledge that Joe went to a restaurant and left satisfied, an application could answer questions such as:

Did Joe eat?
Did he pay for a meal?
Did he tip the waiter or waitress?

2.3.5 Procedural Methods

AI systems utilize both procedural and declarative forms of knowledge. Declarative knowledge specifies what needs to be done while procedural knowledge is most commonly used to specify how something is to be done. Higher level AI languages shield the user from as much of the procedural part as possible by supplying inference mechanisms which operate on user supplied declarative knowledge in a certain format. However, it is well recognized that considerable efficiency can be gained in large systems by allowing the user to specify some of this procedural information. In the language, PLANNER, procedural knowledge is supplied by stating the sequence of subgoals for satisfying a goal. Such heuristics simply specifiy efficient plans for satisfying a localized goal within the problem domain. PLANNER uses simple backtracking and these local goal heuristics to satisfy larger goals which can be presented to the system by the user. However, the price paid for this added flexibility is the loss of modularity and monotonicity.

Another example of a procedural method of representing knowledge is called procedural attachment and is commonly used in frame representations of knowledge. If we consider a person frame in which a specific individual has a specific name, height, hair color and birthdate. Even though we want age to be a slot within this frame, it would be accurate only temporarily if specified as a value. In this case, frame systems which allow procedural attachment would accept a procedure or function which would determine the correct value when needed.

2.3.6 Analogical and Direct Methods

Analogical representation schemes (or direct methods as they are sometimes called) are an attempt to store visual information in a more compact and more meaningful form. The amount of data contained in a diagram such as a map or a figure representing a geometry problem can often be enormous, and scanning a conventional data structure to find other objects can be very time consuming. An analogical representation typically stores a diagram such as a map in a two-dimensional array using the array indexing structure to represent the positions of objects on a map. Not only is this efficient in terms of storage, but it also makes the task of finding neighboring objects far simpler. Notions of distance, complex winding paths, barriers, and the like are also aided by such a representation.

Some work has also been done which suggests that more abstract uses of direct representations are possible. For any application in which proximity is important, such as a chess board or even in a semantic network, it is possible to represent subtleties which might otherwise be impractical.

Of course, there are drawbacks: such representations require special inference mechanisms, custom-tailored to each situation. Also, they often cannot deal with incomplete or indirect information. For example, if it is only known that an object is equidistant from two cities on the map, where should it be represented in the array? Nevertheless, the power of such representations will make them important in future work in AI especially in light of research which suggests a close resemblance to the human cognitive process.

2.4 KNOWLEDGE BASE MANAGEMENT SYSTEMS

Other researchers and even some companies are exploring the possibilities of general purpose systems for managing knowledge. Such systems would attempt to deal with knowledge as a data base management system deals with data. The object is to provide to users of commonly available computer system a powerful tool which can accept knowledge in the most natural forms and perform inferencing over that knowledge base in response to a query.

Expert system building tools such as EMYCIN (Empty MYCIN) or KAS (Knowledge Acquisition System) provide a framework for managing knowledge but have certain limitations. Knowledge must be meticulously coded usually in the form of production rules. Furthermore, the particular inferencing mechanism used by that system may or may not be appropriate for the problem at hand. Such tools are usually limited to special computer architectures which have been widely used for AI reserach but have not found their way into common usage.

Although there is no real agreement as to what constitutes a KBMS, certain themes emerge more often than others. Natural language interfaces, use of formal logic for knowledge representation and notions of modal and temporal data storage and retrieval are common topics. Somewhat more obscure but equally fascinating are such ideas as embedded expert systems for determining data location within distributed data bases [Jarke 83] and automatic generation of data base schemata. All of these approaches seek to make use of data base management systems in the development of knowledge base management concepts.

Since DBMS technology and knowledge-based system technology have advanced separately, the task of combining their strengths into a single package presents no small undertaking. One rather modest approach is to extend the relational data model to include temporal logic [Clifford 83] in support of the historical data base concept. In this model, entries would never be deleted from the data base; instead, both current and past data are retained so that queries can be posed regarding changes to the data over time. More ambitious data models propose that additional semantics and data types such as matrices, time series, ordered sets, and vectors be included in the data definition language and that the operations over these complex objects be included as part of the data manipulation language [Su 83].

The most significant step away from conventional data base technology is the inclusion of an inference capability. This can be achieved in a number of ways. A number of researchers have suggested an extension of the relational data model by replacing the user interface with a logic level language such as first-order logic. This would provide the capability to store and manipulate knowledge and retrieve facts interactively from the knowledge/data base.

In contrast, the KM-1 project at System Development Corp [Kellogg 82] retains a separation between the DBMS and the KBM by interfacing a XEROX 1100 with a Britton-Lee IDM-500. The Xerox 1100 handles the KBM function in the form of a LISP program and the relational data base machine is interfaced through hardware. Queries are submitted to the XEROX in an English-like form where they are transformed into a logical formalism. The KBM constructs a plan by means of the necessary search and inference components using the data base machine to do the searching.

3.0 FUNCTIONALITY OF A KNOWLEDGE BASE MANAGEMENT SYSTEM

Ideally a Knowledge Base Management System should allow a user to enter data and knowledge about what the data means in a form meaningful to the user, and he should be able to pose queries in plain English and receive "intelligent" answers. Of course, the information in the KBMS must be consistent and complete, therefore if contradictory knowledge is entered, a response indicating inconsistent or incomplete data would be issued.

Such a KBMS may not materialize until far into the future, but much has been done already to support these goals. Indeed, the Japanese have taken on this task, at least in part, with their Fifth Generation Computer Project. This section will elaborate on the functionality of such a KBMS within the framework of current research.

3.1 KNOWLEDGE REPRESENTATION

The ideal KBMS would allow a user to enter data and knowledge in various forms, stating the degree of certainty which he feels about the truth or validity of this knowledge. Then when requested, the system would be able to utilize this knowledge to give intelligent answers much as a human does. However, this would require a universal knowledge representation and an extremely powerful inference mechanism. Even though research has not solved these problems, many possibilities are discussed in this section.

3.1.1 Multiple Representations of Knowledge

The semantic network and procedural knowledge models (sometimes called ad hoc) lack the formal, theoretical basis of logic models. It is claimed that the lack of formalism is the reason that they are more efficient !Rich 801. Since much of AI centers around selecting an efficient knowledge representation for a particular application, a knowledge base management system might accommodate custom inference engines. This metaknowledge could be stored in the data dictionary along with data and knowledge descriptions.

In a typical AI application, the inference engine and knowledge representation act in concert to exhibit an intelligent behavior. Between these partners is a kind of filter, which may be thought of as operations or methods. The closer this filter or interface is moved to the inference engine side, the higher the level of abstraction, which means an easier application implementation. The goal of a KBMS should be to handle semantic nets (structured networks like conceptual dependencies, frames, and scripts), procedural methods, production rules, and logic as efficiently as a modern DBMS handles data.

3.1.2 Certainty Factors

In all areas of knowledge, there is some degree of uncertainty regarding the truth of propositions. When we evaluate a situation, we consider the source of information and either consciously or unconsciously place a confidence factor on the various assertions before coming to any conclusions. Even then we may assign a confidence factor to the conclusion which will likely just reflect strong or weak feeling about its validity.

In dealing with the real world, certainty factors are unavoidable. Unfortunately, there is no precise mathematical basis for combining these factors in order to arrive at the appropriate certainty of conclusions. This has not stopped researchers from building systems which make use of such schemes. The expert system MYCIN, is one example. These systems handle uncertainty differently and each uses somewhat "homespun" logic for assigning confidence factors to their conclusions. MYCIN uses a scale from -1 to +1 where -1 is to be interpreted as a complete lack of confidence. When MYCIN is trying to determine the appropriate treatment for an unknown infection, MYCIN may ask "Did the organism grow aerobically?", to which the physician may answer "Yes (.8)". This would indicate that the physician was reasonably sure that the test was positive.

The use of certainty factors has many potential benefits. When conflicting data is encountered, it is possible to make decisions by eliminating the weaker proposition. Another benefit is that search trees can easily be pruned based on low certainty, analogous to the way a human eliminates weak or unlikely possibilities.

3.1.3 Metaknowledge Representation

Metaknowledge is simply knowledge about knowledge. It is used by the system to guide its efforts to interpret data and make inferences. It may be as direct as to provide an axiom for proving some assertion or it may be as obscure as a "cut" in a PROLOG program which tells the system to stop backtracking and go on to the next statement. The separation of domain knowledge from control knowledge has been a controversial issue in AI, but it is generally accepted that for large knowledge bases some improvements will be needed for the knowledge base manager to readily determine the appropriateness and consistency of various control information used by large knowledge based systems of the future.

Even if one subscribes to the extreme position that the inference mechanism in the KBMS should contain all the "brains" and should be independent of control information from the user, there should be no argument against considering "suggestions" from the user which could help the efficiency of the search, provided they do not obsuscate the built in "common sense" of the system. Analogously, even the brightest person may have a great deal of difficulty interpreting facts within an unfamiliar problem domain until certain knowledge of "navigation" is learned from having worked within the field.

An important question is how to represent such metaknowledge so that it is both usable to the system and is maintainable by the user, but much work remains to be done in this area before we will have any solid answers. MRS separates control knowledge from the rest of the program and has had success in reducing execution time for certain problems. TEIRESIAS relies heavily on metaknowledge in questioning experts in the construction of knowledge bases. The importance of such knowledge within a KBMS cannot be overestimated when one considers how closely it represents "deeper" forms of human knowledge.

3.2 TRANSFER OF KNOWLEDGE FROM EXPERT TO KNOWLEDGE BASE

When the knowledge engineer has selected an appropriate knowledge representation and a method to utilize the knowledge (the inference engine), then the task of knowledge acquisition begins. First, a willing expert must be found. Next, is the hurdle of getting the expert to reveal the facts and heuristics used to solve problems. Often, the problem domain lacks complete laws and theories. Quite often, the expert knows there are times when he should break the rules, but cannot explain the details for breaking rules. At this point, the process becomes more complicated. The expert starts to change his mind about how he solves problems. The knowledge engineer begins to redesign the knowledge representation and inference engine. Nevertheless, this dynamic process can result in an expert system that is quite impressive, often matching or exceeding the performance of the human expert [Feig 82].

3.3 KNOWLEDGE MAINTENANCE

This section discusses capturing knowledge and integrating it with existing knowledge in a manner that is consistent with the knowledge already in place. Among the issues of knowledge maintenance are how to provide adequate tools for creating, modifying and discarding knowledge while maintaining consistency and completeness. Adequate tools for preventing errors and enhancing the quality of knowledge will be indispensable in a KBMS.

3.3.1 Create, Modify, Discard

Beyond the normal data management issues of creation, modification and deletion (e.g. concurrency control, transaction management) knowledge maintenance has some special requirements in this regard. Whether or not a record should be inserted into a typical data base does not normally depend upon what the other records contain. On the other hand, rules by their very nature interact. Whether or not a rule should be inserted depends greatly on what rules already exist (see the next section.) Consequently, the knowledge base manager will require special tools for understanding the effects of rule insertion, modification and deletion. For example, a mechanism that isolates the rules which have a connection with the particular rule in question would be invaluable.

3.3.2 Consistency Checks

A simple case of consistency checking is: if both "A implies B" and "B implies C" are in the knowledge base, then the entry "A implies not C" would be inconsistent and should not be allowed in the knowledge base. This particular example of checking is relatively easy to implement, but the problem becomes much more complicated when knowledge is included for situations breaking rules. (Experts know when to break rules).

Creating, modifying, or discarding entries in a knowledge base can become complicated because the effect of a change to a specific fact or rule may be difficult to anticipate. Modification of just one entry could deactivate a large portion of the knowledge.

Adding rules to an expert system may cause unwanted rules or block access to valid rules. Investigating how rules interact and the effect of changes to rules by other rules must be a major thrust in KBMS design research. Perhaps standard methods for handling these and other problems can be developed.

3.3.3 Completeness Checks

The purpose of a completeness check is to discover if any fact or rule is missing from the knowledge base. The form of the completeness check depends upon the knowledge representation at hand. In the case of a rule-based system such a check would identify rules that could never be fired because no combination of other rules can establish the necessary input conditions. In a frame-based system, references to missing frames could be detected or it may be determined that a particular instantiation of a frame has an unreasonable number of empty slots. For semantic nets, a completeness check might identify nodes which seem to have arcs missing. Such mechanisms would require considerable research and development.

3.4 KNOWLEDGE UTILIZATION

"Knowledgeable behavior" is the purpose of a knowledge base management system. This requires a data structure containing knowledge and data and a mechanism that infers new knowledge or data directed toward the answer or goal of a given query. This intelligent utilization of the knowledge will likely make use of such things as the data dicitionary, logical inference rules, fuzzy set theory, and even heuristic rules which provide "hints" about how to navigate efficiently within the knowledge base.

3.4.1 Inference Capability

In order to provide the most flexibility to the user, the KBMS should be capable of "deep" reasoning, bordering on "common sense" which can be augmented by user-supplied heuristics which help explain the idiosyncracies of the specific problem domain.

Along the way to a high-level general reasoning capability, there are many plateaus. The use of simple deductive schemes which can manage monotonic knowledge representations such as production rule bases and first-order logic should certainly be the first step. Heuristics to guide the search for inference candidates for reasons of efficiency could follow. These should be maintained in a separate area of the knowledge base so as not to confuse the facts concerning the problem with the facts which are supposed to help the inference mechanism do its job. This makes the explanation and justification task easier. Further, if the heuristics are inconsistent or cumbersome, they could be ignored by the inference mechanism since they have been kept separate from the knowledge domain facts.

The next step would probably take the form of fuzzy set theory. This allows rules to use the inexact but not necessarily unquantifiable modifiers such as "high" and "low." For example, the rule "Recalculate the air conditioning load when a high-power item is added to the compartment" is a "reasonable" rule provided that "high" can be deducted from other rules which appraise the current context.

In order to reach the plateau of deep knowledge, it will require a highly sophisticated mechanism which will be able to handle nonmonotonic knowledge, ambiguity, redundancy, and other irregularities in an intelligent way. This will show itself in the ability to degrade gracefully as does a human when things get muddled. It will be necessary to keep track of all the conflicting data and make excuses and explanations about its inability to make a decision based on the inconsistencies.

3.4.2 Handling Uncertainty

In a number of expert systems, rules can be expressed using certainty factors. The inferencing mechanism is required to deal with these measures of uncertainty and draw reasonable conclusions. Since there is no mathematical basis for assigning certainty factors to possible conclusions which can be inferred from such rules, a kind of seat of the pants logic is applied which simply "works". In the MYCIN system, this mechanism is called "inexact logic" and works like this. Certainty factors have values between -1 and +1. Complete certainty is represented by +1, and "I don't know" is represented by zero, and complete confidence that it won't happen is represented by -1. A rule such as "If (A and B) then C(0.6)" means that if we know A and B to be true, then we can assert C with a certainty of +0.6. The heuristic for summing certainty factors is simple. If we know A with a certainty of +0.3 and we know B with a certainty of +0.7, then we can asert (A and B) with a certainty of

+0.3 or the minimum of the two certainties. In the case of an OR conditions, the maximum of the two values is used. The implication operation is simple multiplication of the resultant certainty on the left side with the certainty on the right side of the rule. Thus, the certainty of C in our example above can now be established as +0.18. Of course, the closer we are to zero, the less sure we are about the conclusion. It is easy to see how a low certainty factor will not allow much propagation of other facts and consequently forces the knowledge engineer to select certainty factors for his rules which will propogate appropriate certainty factors to related rules. In other words, the knowledge engineer has to retain a keen sensitivity to the arithmetic of the system. There is really no way to prove that the arithmetic is appropriate. It will work provided that the knowledge engineer selected "good" values.

Representing knowledge would be much simpler if certainty was not an issue. Since this is not the case and since we can be assured that a very large rule bases will be required for future systems, we surely will need some improvements which will make the knowledge engineering job easier. An analysis tool which would analyze the rule base and pinpoint "weak" rules, inconsistencies, and other helpful observations could be one approach. Another option might be to invent a whole new language of certainty factors with a new arithmetic. It may even be quantified in "fuzzy" terms such as "high" and "low" and the arithmetic may be nonlinear.

3.4.3 Dynamic Knowledge Acquisiton and Validation

During the execution of an AI application, knowledge will change because of new facts or rules from outside the system or those discovered by the inferencing process. If the new knowledge must be placed in the knowledge base, all validation of sections 3.1.1 and 3.1.2 must be performed at the time of update.

3.4.4 Explanation and Justification

As knowledge is accessed by the system, a log must be kept of its logical progression toward the goal in order to satisfy a request for explanation or justification from the user. A dialog like the following might even be possible.

User: Suggest modification.

System: Use a lighter part at Point X.

User: Why?

System: 1) Our goal is to design a light, safe aircraft.

2) The part at Point X exceeds safety margin by a factor of 1.3.

3) A lighter part will still meet safety requirements and save weight.

This gives the user confidence in the system as well as supplying a method of spot evaluation of system accuracy. Such a facility could also serve as a training tool for transferring expertise to novices.

3.4.5 Multiple Users

One advantage of a KBMS compared to current AI applications would be that a central knowledge source could be shared by many applications. Conventional DBMS techniques could be used for locking and transaction processing. This feature would obviously save time developing various knowledge base applications, but more importantly will provide a central, consistent repository for knoweldge, allowing large systems to be constructed in a manageable and maintainable way.

3.4.6 Multiple Perspectives

Once the basic knowledge representations are selected for a knowledge base application, the view of these representations can be controlled by schemas for different perspectives. For example, an engineering knowledge base for an airplane wing could be constructed as a semantic network. Then frames and scripts could be employed to provide different perspectives of the underlying wing knowledge. Various perspectives could describe the wing in different terms, depending on the application. The different representations might be: a script describing the steps in structural analysis, a frame representing the wing in terms of its stress components, and a frame describing vibration.

The system could construct the necessary mappings automatically in order to support these various views similiar to current DBMS systems.

3.4.7 Flexible Multipurpose User Interface

The knowledge manager, application designer, and end user will have different roles and needs for viewing and updating the knowledge base.

The knowledge manager will be responsible for transferring human knowledge into the system and will need flexible, natural interfaces for this task. In addition, functions which display the effect of modifications to the knowledge base will be of great value (see 3.3.1). The application designer will need a high-level, application interface in order to produce effective and usable products. In addition to knowledge management functions, the application designer may also want compatibility with current data base design technology. This can facilitate the evolution to knowledge base management without having to redesign much of the current body of software. End users will also need powerful, innovative access to the knowledge base, perhaps in the form of natural language.

3.4.8 Security, Backup, and Recovery

Recovery mechanisms are needed to restore data and knowledge to a known consistent state following a hardware or software failure. In conventional data base systems, recovery is accomplished using backup copies and logs. The data base is restored by using the log to redo or undo transactions as required. It seems reasonable that these techniques will be adequate when applied to the knowledge base architecture.

BIBLIOGRAPHY

[Barr 81] Barr, A. and Feigenbaum, E. A., "The Handling of Artificial Intelligence," HeurisTech Press, Stanford, California 1981.

[Clifford 83] Clifford, J. and Warren, D. S., "Formal Semantics for Time in Databases," ACM Transactions on Database Systems, 6.2, June 1983.

[Feig 82] Feigenbaum, E. A., "Knowledge Engineering for the 1980's, "Computer Science Sept., Stanford University, 1982.

[Gevarter 83] Gevarter, W. B., "An Overview of Artificial Intelligence and Robotics," NASA Technical Memorandum 85838, 1983.

[Hofstadter 80] Hofstadter, Douglas R., "Godel, Escher, Bach, An Eternal Golden Braid," Random House, New York, New York 1980.

[ICOT 82] Kunifuji, S. and Yokota, H., "PROLOG and Relational Data Bases for Fifth Generation Computer Sciences," ICOT Technical Report TR-002, Institute for New Generation Computer Technology, Tokyo, Japan, 1982

[Israel 83] Israel, Bolt Beranek and Newman, Inc., "The Role of Logic in Knowledge Representation," Computer, October 1983 issue.

[Jarke 83] Jarke, M. and Vassiliou, Y., "Coupling Expert Systems with Database Management Systems," Proc. NYU Symposium on Artificial Intelligence Applications to Business, New York, 1983.

(Kellog 82] Kellogg,, C., "A Pratical Amalgam of Knowledge and Data Base Technology," AAAI, 1982, pp. 306-309.

[Rich 83] Rich, Elaine, "Artifical Intelligence", McGraw-Hill, 1983.

[Su 83] Su, S. Y. W., "A Semantic Association Model for Corporate and Scientific-Statistical Databases, "Information Sciences 29, 1983.